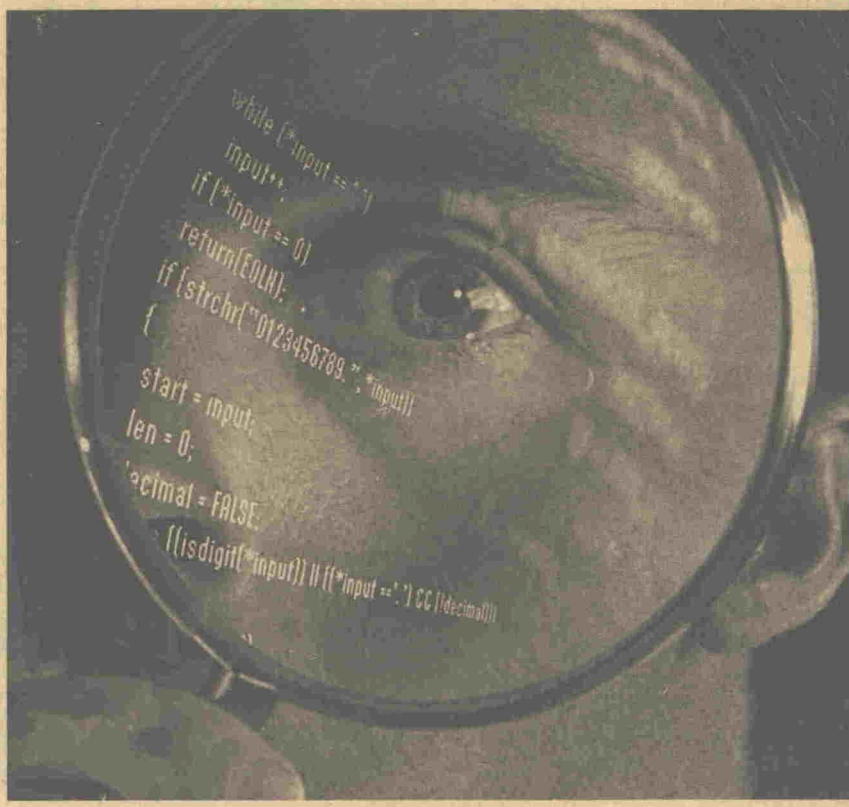
อนุชาติ ลาสนธิ

SOFT MAIL

โปรแกรมวิเคราะห์รหัสผ่าน
หน่วยสืบราชการลับ

ไม่ว่าใครจะใช้มาตรการป้องกันข้อมูลด้วยรหัสผ่าน เราก็สามารถสอดแนมเข้าไปค้นหารหัสผ่านนั้นได้ แต่อย่าถึงขั้นร้ายแรงนัก



เมื่อพูดถึงระบบความปลอดภัยของแต่ละระบบคงจะไม่เหมือนกัน ถ้าเราคุยกับเสี่ยธนาคารเรื่องความปลอดภัย *"สบายมาก ปลอกภัยร้อยเปอร์เซ็ง ตู้เซ็กของเราใช้เหล็กหนาตั้ง 4 นิ้ว"* แต่อย่าเผลอไปถามเรื่องนี้กับคุณสุภาพสตรีเข้าล่ะ ไม่งั้นอาจจะได้คำตอบทำนองที่ว่า *"กระซับ แห้งสบาย ไม่หวั่นแม้วันมามาก"* นะจะบอกให้ และที่ขาดเสียไม่ได้ที่จะต้องกล่าวถึง *"ระบบความปลอดภัยของเราได้มาตรฐานมีวาล์วควบคุม 4 ตัว หากตัวใดตัวหนึ่งขัดข้องหรือรั่ว ตัวที่เหลือจะทำงานแทนทันทีครับ ไม่มีการระเบิด (แก๊ส) แน่"* แต่อย่าให้เรื่องนี้เข้าถึงหูโปรแกรมเมอร์ก็แล้วกันไม่งั้น *"ระบบความปลอดภัยข้อมูลของเรามีการถามรหัสผ่าน 3 ครั้ง แต่ละครั้งต้องใส่รหัสผ่านถึง 9 ตัวในเวลา 4 วินาที"*

โอ้โฮ! ฟัง ๆ ดูแล้วทุกคนต้องการความปลอดภัยกันทั้งนั้น บทความนี้เป็นบทความที่เกี่ยวกับทั้งความปลอดภัยและไม่ปลอดภัยของข้อมูลโดยใช้หลักการกดรหัสผ่าน เพื่อจะเข้าไปหาข้อมูลหรือโปรแกรม

เจาะกึ๋นรหัสผ่าน

ระบบรักษาความปลอดภัยของคอมพิวเตอร์เพื่อปกป้องข้อมูลหรือโปรแกรมส่วนมากแล้วจะใช้วิธีป้อนรหัสผ่าน ถ้าเราสามารถกดรหัสผ่านได้ถูกต้อง เราก็สามารถเข้าถึงส่วนอื่น ๆ ของโปรแกรมต่อไปได้ หากตอบไม่ถูกโปรแกรมจะหยุดชะงักทันที หรืออาจทำการรีเซตเครื่อง เพื่อป้องกันไม่ให้ผู้บุกรุกเข้าไปยุ่งกับโปรแกรมหรือข้อมูลนั้นได้ รหัสผ่านที่ว่านี้ไม่ต้องยาวมากหรอกค แค่สองตัวก็ยังไม่รู้เลยว่าใช้เวลาทั้งว จะกดถูกหรือเปล่า แต่มีข้อสังเก อีกเล็กน้อยคือ คนที่ตั้งรหัสผ่านจะตั้ กำหนดในสิ่งที่ตัวเองจำได้ง่าย ๆ อย่างเ เบอร์โทรศัพท์ ชื่อ นามสกุล วันเ หรือชื่อแฟน ฯลฯ นักวิเคราะห์รหัสผ่ ทั้งหลายคงพอจะเดาถูกได้บ้าง ถ้าลองทั้งสี่ห้าอย่างแล้วยังไม่สำเ ก็ถึงคราวที่จะต้องงัดเอาไม้ตายมา แล้วล่ะ

กระบวนท่า FP.COM คืออะไร

FP.COM คือโปรแกรมที่จะช่วยให้เราวิเคราะห์รหัสผ่านของโปรแกรมนั้นได้ วิเคราะห์ยังไง? หลักการของโปรแกรม FP.COM มีอยู่ว่า เมื่อรันโปรแกรมแล้ว ต่อไปทุก ๆ คีย์ที่กดเข้ามาจะถูก FP.COM เก็บไว้หมด สามารถเรียกมาดูภายหลังได้ การเรียกโปรแกรม FP.COM ทำได้สองลักษณะ คือ เรียกโดยไม่ระบุพารามิเตอร์ตามหลัง การทำงานจะเหมือนกับที่กล่าวข้างต้น วิธีที่สองเป็นการเรียกโดยมีพารามิเตอร์ตามหลัง พารามิเตอร์ที่ใช้คือเลข 1-9 เช่น FP/5 เป็นต้น พารามิเตอร์ที่ต่อท้ายนี้หมายถึงค่าเวลาที่ตั้งไว้ (Timer มีหน่วยเป็นนาที) ก่อนที่จะเริ่มเก็บ หมายความว่าเมื่อเรียกโปรแกรม FP แล้ว จะยังไม่มีการเก็บคีย์ทันที จนกระทั่งถึงเวลาที่ตั้งไว้จึงจะเริ่มเก็บ เช่น FP/3 เวลาผ่านไป 3 นาทีจึงจะเริ่มเก็บคีย์ที่กดเข้ามา

แน่นอนที่สุดการที่โปรแกรม FP.COM จะสามารถช่วยวิเคราะห์รหัสผ่านออกมาได้นั้นเราจะต้องให้คนที่รู้รหัสผ่าน (แต่ไม่ยอมเปิดเผย) มาใช้เครื่องก่อนโดยที่เราต้องรันโปรแกรม FP.COM รอไว้แล้ว ต่อไปนี้เป็นเหตุการณ์สมมติของการล่อเหยื่อมาติดกับ

"โจ๊ก นายช่วยเรียกโปรแกรมของนายออกมาให้เราดูหน่อยสิ"

"มากไปแล้วนายหยวก ได้ไง ๆ อย่างงี้นายก็รู้รหัสผ่านของเราสิ"

"แค่รหัสผ่านเราไม่ดูหรอกน่า (เพราะเดี๋ยวก็รู้) *อยากเห็นหน้าตาโปรแกรมของนายเท่านั้นแหละ"* และถ้านายหยวกพูดกล่อมได้สำเร็จ นายโจ๊กจะเดินมาเคาะคีย์บอร์ดสามสี่ห้าคีย์ เมื่อถึงขั้นตอนกดรหัสผ่าน

"นายหันหน้าไปทางอื่นสิหยวก เราจะกดรหัสผ่านแล้ว" (ระแวงซะจริง ๆ เชียว) รหัสผ่านถูกกดเข้าไปเรียบร้อย รอสักครู่หนึ่งโปรแกรมนั้นจึงยอมให้ใช้งาน

"เออ! พอแล้ว ๆ ขอดูแค่นี้แหละขอบใจมาก" (ตรงตามแผนทุกอย่าง) เป็นอันว่าเสร็จสิ้นขั้นตอน 'จารกรรม' รหัสผ่านคราวนี้เมื่อนายโจ๊กเผลอ นายหยวกจะกดคีย์ตัวหนึ่ง ตัวอักษรทุกตัวที่นายโจ๊กพิมพ์เข้าไปจะแสดงให้ดูที่หน้าจอ โดยรวมถึงรหัสผ่านของนายโจ๊กด้วย

เมื่อหมดภาระของโปรแกรม FP.COM แล้ว ต่อไปก็เป็นหน้าที่ของเราเองที่จะวิเคราะห์หรือเดาว่า คีย์ที่กดเข้ามาตัวไหนบ้างที่มีหน้าตาเหมือนรหัสผ่าน ซึ่งเป็นเรื่องที่....หวานหมูมากมิใช่หรือ

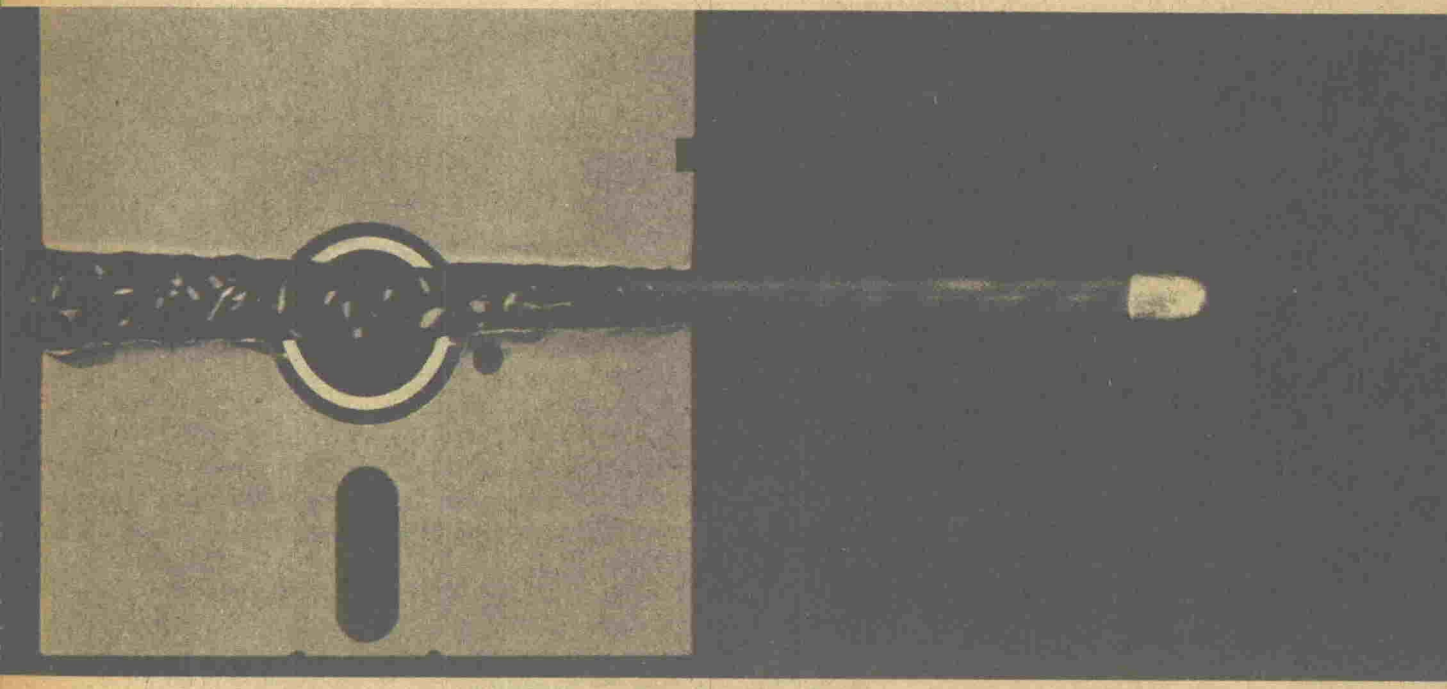
ทำความเข้าใจโปรแกรม

ในโปรแกรม FP.COM มีการกำหนด Equ ไว้ 4 ตัว

Video : เป็นแอดเดรสหรือหน่วยความจำของจอภาพ ถ้าเราใช้จอสีต้องเปลี่ยนจาก 0B000H เป็น 0B800H

Look_key : เป็นสแกนโค้ดของคีย์ที่ใช้สำหรับดูค่าที่เก็บไว้ เพื่อเดาว่าตัวไหนน่าจะเป็นรหัสผ่าน ในโปรแกรมกำหนดไว้เป็น 104 ซึ่งก็คือคีย์ Alt-F1 นั่นเอง

Clear_key : คีย์ที่เก็บเข้ามาเพื่อใช้วิเคราะห์หารหัสผ่านนั้นจะเก็บไว้ในบัฟเฟอร์ที่ตั้งขึ้นมาเองในโปรแกรม ชื่อบัฟเฟอร์มีขนาด 400 ตัวอักษร ดังนั้นเมื่อเก็บจนครบ 400 ตัวอักษรหรือ 400 คีย์ก็จะหยุดเก็บอัตโนมัติ โดย



ปล่อยผ่านไปปกติ แต่ในกรณีที่เหยื่อของเราไม่ยอมกดรหัสผ่านสักที มัวไปทำโน่นทำนี่อยู่ ถ้าเรากลัวว่าบัฟเฟอร์จะเต็มเสียก่อนก็ต้องชิงกดคีย์นี้เพื่อทำการล้าง (Clear) บัฟเฟอร์ การเก็บจะเริ่มต้นที่ตัวแรก ซึ่งสแกนโค้ดของคีย์ในโปรแกรมกำหนดไว้เป็น 113 คือ Alt+F10

Clear_time : ในกรณีที่เรียกโดยมีพารามิเตอร์ตามมา หรือมีการตั้งเวลาไว้ล่วงหน้า เช่น FP/5 เมื่อรันโปรแกรมผ่านไปแล้ว 5 นาทีจึงค่อยทำการเก็บคีย์ แต่ถ้าจวนจะถึงเวลาที่ตั้งไว้อยู่แล้ว เหยื่อของเรายังไม่โผล่หน้ามาเลย เราก็สามารถกดคีย์ตัวนี้เพื่อให้เริ่มนับใหม่ ในโปรแกรมกำหนดเป็น 108 คือ Alt+F5

Clear_delay : เป็นตัวสุดท้ายที่ควรทำความรู้จัก ในกรณีที่เรียกโปรแกรมโดยมีพารามิเตอร์ตามมาเพื่อตั้งเวลานั้น หากเราเกิดเปลี่ยนใจไม่อยากตั้งเวลาแล้ว พอดีกับที่เหยื่อกำลังเดินมาแต่เวลาที่ตั้งไว้เหลืออีกตั้งนาน ก็ให้กดคีย์ตัวนี้เพื่อยกเลิกการตั้งเวลาแบบถาวร หมายความว่าเมื่อเรากดคีย์นี้แล้วมากดคีย์ Clear_time (Alt+F5) ภายหลังจะไม่มีการตั้งเวลาใหม่ คีย์ตัวนี้ในโปรแกรมกำหนดไว้เป็น 111 คือ Alt+F8

สแกนโค้ดทั้ง 4 ตัวผู้ใช้สามารถเปลี่ยนได้ตามใจชอบ

ส่วนตัวแปรสำคัญในโปรแกรมที่น่ารู้มีอยู่ 2 ตัว ตัวแรกชื่อ Buffer และตัวที่สองชื่อว่า Video_buf ทั้งสองตัวนี้มีความสัมพันธ์กัน กล่าวคือ Video_buf เป็นหน่วยความจำที่จองไว้สำหรับเก็บจอภาพบรรทัดที่ 1-8 (ในหนึ่งบรรทัดใช้หน่วยความจำ 160 ไบต์ ผู้เขียนจึงกำหนดไว้เป็น 8×160 ทำไมต้องเก็บไว้ด้วย? ในกรณีที่เรากด Look_key (Alt+F1) คีย์ที่เก็บไว้จะแสดงออกมาทับบรรทัดที่ 1-5 เพราะว่าในโปรแกรมกำหนดบัฟเฟอร์สำหรับเก็บคีย์ไว้ 400 คีย์ (5×80 = 400) เมื่อยกเลิกการแสดงคีย์ที่เก็บไว้ให้ดูแล้ว จะต้องคืนข้อความเดิมของบรรทัดที่ 1-5 ให้เป็นเหมือนเก่านั่นเอง

ถ้าเราต้องการเปลี่ยนค่าสองตัวนี้จะให้ค่าบัฟเฟอร์มากกว่า Video_buf/2 ไม่ได้ แต่ต้องให้ค่าน้อยกว่าหรือเท่ากัน ในโปรแกรมบัฟเฟอร์กำหนดไว้เป็น 400 หรือ 5 บรรทัด และ Video_buf กำหนดเป็น 8×160 หรือ 8 บรรทัด ถ้าคิดว่า 400 ตัวไม่พอเก็บเราก็เพิ่มค่าของบัฟเฟอร์ได้อีก 3 บรรทัด หรือเพิ่มความสามารถในการเก็บอีกจาก 240 เป็น 640 คีย์โดยไม่ต้องเพิ่มค่า Video_buf เลย

โปรแกรมหลัก FP.COM จะมีโปรแกรมหลักอยู่ 2 โปรแกรมคือ

- Password Proc Far เป็นโปรแกรมที่สร้างขึ้นมาใหม่แทนที่โปรแกรมเก่าของ INT 16H พูดง่าย ๆ ก็คือ เมื่อรัน FP.COM แล้วทุกครั้งที่มีการกดคีย์ซีพียูต้องกระโดดมาทำงานที่นี่ ในส่วนนี้ลองตรวจสอบคีย์ทั้ง 4 คีย์ ถ้า

ใช่ก็ทำงานตามหน้าที่ของคีย์นั้น เช่น ถ้าผู้ใช้กด Alt+F1 ที่จอภาพจะแสดงคีย์ที่ผู้ใช้กดผ่านมาทั้งหมดให้ดู

• Check_time Proc Far เป็นโปรแกรมตั้งเวลาในกรณีที่เรียกแล้วตามด้วยพารามิเตอร์ แต่ถ้าเรียกโดยไม่มีพารามิเตอร์โปรแกรมส่วนนี้จะไม่ทำงาน

โปรแกรมย่อยทุกโปรแกรมจะถูกเรียกใช้โดยโปรแกรมหลักพาสเวิร์ด

• Cls_buf ล้างหรือเคลียร์บัฟเฟอร์ให้ว่างให้กรณีที่กดคีย์ Alt+F10

• Save_scr เก็บจอภาพจากบรรทัดที่ 1-8 ไว้ในตัวแปรชื่อ Video_buf

• Write_pass นำค่าที่เก็บในบัฟเฟอร์หรือคีย์ที่เก็บไว้มาแสดงที่หน้าจอ

• Get_key หลังจากที่แสดงให้ดูแล้วจะหยุดรอการกดคีย์หนึ่งครั้งเพื่อกลับสู่สภาวะปกติ

• Restore คืนจอภาพที่เก็บจากบรรทัดที่ 1-8 ให้เหมือนเดิม

เมื่อพิมพ์โปรแกรมเรียบร้อยแล้วให้ตั้งชื่อว่า FP.ASM ซึ่งเป็นภาษาแอสเซมบลีมาผ่านขบวนการแปลงให้เป็นภาษาเครื่องหรือที่เรียกกันว่าการแอสเซมเบลอร์ โดยใช้โปรแกรมมาโครแอสเซมเบลอร์ (MASM.EXE) ถ้าคีย์โปรแกรมไม่ผิดการแอสเซมเบลอร์จะเป็นดังรูปที่ 1 และจะได้ไฟล์ชื่อว่า FP.OBJ นำไปลิงก์ต่อโดยใช้ LINK.EXE จะได้ FP.EXE เพิ่มขึ้นอีกไฟล์หนึ่ง หากยังรันไม่ได้เพราะโปรแกรม FP.ASM เขียนในลักษณะของโปรแกรมนามสกุล .COM จึงต้องเปลี่ยนจาก .EXE เป็น .COM ก่อนโดยใช้โปรแกรม EXE2BIN. EXE บางคนที่ไม่คุ้นเคยอาจถามว่าใช้คำสั่ง REN เปลี่ยนเลยไม่ได้หรือ ทำไมต้องใช้โปรแกรม EXE2BIN ด้วยล่ะ ซึ่งตามหลักการนั้นเราทำไม่ได้ เพราะวิธีการเปลี่ยนไม่เหมือนกัน คำสั่ง REN จะเปลี่ยนแค่นามสกุล แต่โปรแกรม EXE2BIN เปลี่ยนที่ตัวโปรแกรมให้เป็นลักษณะของโปรแกรม .COM จึงจะสามารถรันได้ และโปรแกรม FP.COM เป็นโปรแกรมประเภทฝังหรือเรสซิเดนต์ เมื่อรันแล้วจะอยู่ในหน่วยความจำตลอด

```
C>MASM FP;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985 1987. All rights reserved.

     51568 + 33168 Bytes symbol space free

          0 Warning Errors
          0 Severe  Errors

C>LINK FP;

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021 : no stack segment
C>EXE2BIN FP.EXE FP.COM
```

รูปที่ 1 การแปลงโปรแกรม FP.ASM ให้เป็นภาษาเครื่องโดยใช้โปรแกรมมาโครแอสเซมเบลอร์

สรุปส่งท้าย

เชื่อแน่ว่าผู้อ่านหลาย ๆ ท่านคงจะงงกับการใช้โปรแกรมเหมือนกับผู้เขียนอยู่บ้าง จึงขอสรุปการใช้งานอีกครั้งเพื่อให้เข้าใจยิ่งขึ้น ถ้าเรารันโปรแกรมโดยไม่มีพารามิเตอร์ตามหลังคีย์ที่ใช้จะมีแค่สองคีย์เท่านั้นเองคือ Alt+F1 สำหรับเรียกดูคีย์ที่กดเข้ามา และ Alt+F10 จะล้างคีย์ที่เก็บไว้เพื่อเก็บใหม่ แต่ถ้าเรียกโดยมีพารามิเตอร์ตามมาจะมีคีย์เพิ่มอีกสองคีย์คือ Alt+F5 เริ่มตั้งเวลาใหม่ และ Alt+F8 ยกเลิกการตั้งเวลา ส่วนจำนวนคีย์ที่สามารถเก็บได้ทั้งหมด 400 ตัว เมื่อนึกอยากจะเปลี่ยนก็ทำได้ โดยเราต้องเปลี่ยนทั้ง 2 ค่าคือ Buffer และ Video_buf ค่าของ Buffer จะต้องเท่ากับหรือน้อยกว่าค่าของ Video_buf/2 เช่น เราต้องการเพิ่มความสามารถในการรับคีย์เป็น 500 คีย์ ก็ให้ค่า Buffer = 500 และ Video_buf = 1000 หากผู้อ่านยังไม่เข้าใจอีกผู้เขียนขอแนะนำให้ลองใช้งานเองจะถนัดกว่า

"เมื่อคนไม่ดีมีอำนาจ โลกนี้ย่อมปั่นป่วน" โปรแกรมนี้ก็เช่นเดียวกัน ไม่ควรใช้เพื่อล้วงความลับอะไรบางอย่างในสิ่งที่เจ้าของโปรแกรมหรือข้อมูลนั้น 'หวงสุดขีด' เพราะความวุ่นวายอาจเกิดขึ้นได้ และเมื่อนั้นโลกจะประณามผู้บ่อนทำลายความสงบสุข....แน่นอน

โปรแกรมวิเคราะห์รหัสผ่าน

```
;+---------- Source File ------------------------
;|
;|    ******* FP.ASM *********
;|          Anuchat  Lasonthi
;|           Com 4 Soft Group
;|                Odorn tech
;|            date   10/10/90
;|            Time   17:28:00
;+-----------------------------------------------

Video       Equ 0B000H              ;Video buffer for Monochrome
                                    ;If color change to 0B800H
Look_key    Equ 104                 ;Scan code for look Password (Alt+F1)
Clear_key   Equ 113                 ;Sacn code-for clear buffer  (Alt+F10)
Clear_time  Equ 108                 ;Scan code for clear timer  (Alt+F5)
Clear_delay Equ 111                 ;Scan code for clear delay time (Alt+F8)
;-------------------------------------------------------------------------
Cseg        Segment
            Assume Cs:Cseg,Ds:Cseg
            Org 100H

Start:
            jmp  Install

Old_vect16  DD 00000000             ;Save the interrupt vector (Int 16H)
Old_vect1C  DD 00000000             ;Save the interrupt vector (Int 1CH)
Count       DW 0000                 ;For point to buffer
New_time    DW 0000                 ;Current time+delay time
Status      DB 00                   ;If buffer full then = 0FFH
Stat_time   DB 00                   ;Status of timer
Stat_delay  DB 00                   ;Status for cancel delay time
Parameter   DB 00                   ;Parameter for delay time
Buffer      DB 400   Dup(?)         ;Keyboard Buffer (5 line)
Video_buf   DB 8*160 Dup(?)         ;For save Video Page 0 (8 line)

Mark        DW 1602H                ;For check program already
;-------------------------------------------------------------------------
Password    Proc  Far

            Assume Cs:Cseg,Ds:Cseg
            sti                     ;Reenable interrupts

            cmp ah,00               ;
            jz next_0               ;If Int 16H function 0 jump

            Assume Ds:Nothing       ;/Else
            jmp old_vect16          ;     go back
Next_0:
            push bx                 ;
            push cx                 ;
            push dx                 ;
                                    ;
            push si                 ;
            push di                 ;
                                    ;
            push es                 ;
            push bp                 ;
            push ds                 ;Save altered registers

            push cs                 ;
            pop  ds                 ;.Com files  DS=CS

            pushf                   ;
            call old_vect16         ;Get a character

            push ax                 ;Save  Scan code and ascii code

            cmp stat_delay,0FFH     ;
            jz next_2               ;IF stat_delay 0FFH jump to pass_3
                                    ;(cancel delay time)

            cmp ah,clear_delay      ;IF scan code = clear_delay
            jnz next_1              ;
                                    ;
            mov stat_delay,0FFH     ;Set cancel delay time
Next_1:
            cmp stat_time,00        ;
            jz exit                 ;If stat_time = 00 jump to exit

            cmp ah,clear_time       ;
            jnz next_2              ;If scan code () clear_time jump Pass_3

            mov bx,1092             ;BX = 1 minute (18.2*60)
            mov ah,00               ;
            mov al,parameter        ;
            mul bx                  ;AX = delay time

            push ax                 ;Save old time
            mov ah,00               ;
            int 1AH                 ;Get current time
            pop ax                  ;restore old_time

            add ax,dx               ;Add  N(AX) sec delay
            mov new_time,ax         ;Store in new_time
            mov stat_time,00        ;

            jmp exit
Next_2:
            cmp ah,clear_key        ;
            jnz next_3              ;If scan code () clear_key jump to pass_1

            mov status,00H          ;/Else
            mov count,00            ;Set count = 00
            call cls_buf            ;Clear buffer  and
            jmp exit                ;Jump to exit
Next_3:
            cmp ah,look_key         ;
            jnz next_4              ;If scan code () look_key jump to pass_2

            call save_scr           ;/Else  save screen
            call write_pass         ;Dispaly character in buffer
            call get_key            ;Wait until press a key
            call restore            ;Restore screen
            jmp exit                ;Jump to exit
Next_4:
            cmp status,0FFH         ;
            jz exit                 ;If buffer full then jump to exit

            mov di,offset buffer    ;/Else
            add di,count            ;Update pointer
            mov [di],al             ;Save ascii code
            inc count               ;Increment pointer

            cmp count,length buffer ;
            jnz exit                ;
            mov status,0FFH         ;If buffer full then set status = 0FFH
Exit:
            pop ax                  ;Restore scan code and ascii code
```

```
        pop ds                  ;
        pop bp                  ;
        pop es                  ;
                                ;
        pop di                  ;
        pop si                  ;
                                ;
        pop dx                  ;
        pop cx                  ;
        pop bx                  ;Restore all registers

        iret                    ;End of interrupt

Password    Endp
;-------------------------------------------------------------------
Cls_buf     Proc  Near

        mov di,offset buffer    ;DI point to buffer
        mov cx,length buffer    ;CX = length of buffer
        mov al,32               ;AL = space
Cl1:
        mov [di],al             ;
        inc di                  ;
        loop cl1                ;Clear buffer

        ret
Cls_buf     Endp
;-------------------------------------------------------------------
Save_scr    Proc  Near

        mov ax,video            ;
        mov es,ax               ;ES point to video buffer

        mov cx,length video_buf ;CX = length of video_buf
        mov si,offset video_buf ;SI point to video_buf
        mov di,0000             ;Offset 0000
Sav1:
        mov al,es:[di]          ;
        mov [si],al             ;
        inc di                  ;
        inc si                  ;
        loop sav1               ;Save screen page 0 to video_buf

        ret
Save_scr    Endp
;-------------------------------------------------------------------
Write_pass  Proc  Near

        mov ax,video            ;
        mov es,ax               ;ES point to video buffer

        mov cx,length buffer    ;CX = length of buffer
        mov di,offset buffer    ;DI point to buffer
        mov si,0000             ;SI = offset 0000
        mov bl,78H              ;Reverse attribute
Vr1:
        mov al,[di]             ;
        mov es:[si],al          ;
        inc si                  ;
        mov es:[si],bl          ;
        inc si                  ;
        inc di                  ;
        loop vr1                ;Display character in buffer

        ret
Write_pass  Endp
;-------------------------------------------------------------------
Get_key     Proc  Near

        mov ah,0                ;
        pushf                   ;
        call old_vect16         ;Get a key

        ret
Get_key     Endp
;-------------------------------------------------------------------
Restore     Proc  Near

        mov ax,video            ;
        mov es,ax               ;ES point to video buffer

        mov cx,length video_buf ;CX = length of video_buf
        mov si,offset video_buf ;SI point to video_buf
        mov di,0000             ;DI = offset 0000
Sa1:
        mov al,[si]             ;
        mov es:[di],al          ;
        inc di                  ;
        inc si                  ;
        loop sa1                ;Restore screen from video_buf

        ret
Restore     Endp
;-------------------------------------------------------------------
Check_time  Proc  Far
        Assume  Cs:Cseg,Ds:Cseg
        sti

        push ax                 ;
        push bx                 ;
        push cx                 ;
        push dx                 ;
                                ;
        push si                 ;
        push di                 ;
                                ;
        push es                 ;
        push bp                 ;
        push ds                 ;Save altered registers

        push cs                 ;
        pop  ds                 ;.Com files  DS=CS

        Assume Ds:Nothing       ;
        pushf                   ;
        call old_vect1C         ;Call old INT 1CH
;.....................................................................
        mov ah,00               ;
        int 1AH                 ;Get the time of day count

        cmp dx,new_time         ;Compare reading to delay value
        jl exit_1C              ;If lessthan jump to exit_1C
;.....................................................................
        mov stat_time,0FFH      ;
Exit_1C:
        pop ds                  ;
        pop bp                  ;
```

```
        pop es                  ;
                                ;
        pop di                  ;
        pop si                  ;
                                ;
        pop dx                  ;
        pop cx                  ;
        pop bx                  ;
        pop ax                  ;Restore all registers

        iret                    ;End of interrupt

Check_time  Endp
;-------------------------------------------------------------------
Install:
        push cs                 ;
        pop  ds                 ;DS = CS
;.....................................................................
        mov ah,35H              ;
        mov al,16H              ;
        int 21H                 ;Get the interrupt vector of INT 16H

        mov ax,[bx-2]           ;
        cmp ax,mark             ;
        jnz no_ready            ;If AX = mark ? program already
;.....................................................................
        mov ah,09H              ;
        mov dx,offset ready     ;
        int 21H                 ;Display 'ready'

        mov ah,4CH              ;
        int 21H                 ;Exit to  DOS
;.....................................................................
No_Ready:
        mov word ptr old_vect16,bx    ;Save interrupt vector (IP)
        mov word ptr old_vect16+2,es  ;Save interrupt vector (CS)
;.....................................................................
        mov ax,video            ;
        mov es,ax               ;ES point to video buffer

        mov cx,2000             ;Character per page
        mov di,0                ;DI = offset 0000
        mov ax,0720H            ;Space character and normal attribute
Cls:
        mov es:[di],ax          ;
        inc di                  ;
        inc di                  ;
        loop cls                ;Clear screen
;.....................................................................
        mov ah,02               ;
        mov dx,00               ;
        mov bh,0                ;
        int 10H                 ;Set the cursor to top left

        mov ah,09H              ;
        mov dx,offset show      ;
        int 21H                 ;Display 'Show'
;.....................................................................
        mov di,80H              ;
        mov al,ds:[di]          ;
        cmp al,2                ;
        jnz no_para0            ;Get parameter
```

```
        mov al,ds:[di+2]        ;
        cmp al,'1'              ;
        jl no_para0             ;
        cmp al,'9'              ;
        jg no_para0             ;If parameter <1 or >9 then  jump no_para

        and al,0FH              ;
        mov parameter,al        ;
        mov ah,00               ;Mask out four top bit

        mov bx,1092             ;
        mul bx                  ;BX = 1 minute

        push ax                 ;Save old time
        mov ah,00               ;
        int 1AH                 ;Get current time
        pop ax                  ;restore old time

        add ax,dx               ;
        mov new_time,ax         ;Replace new time
;.....................................................................
        mov ah,35H              ;
        mov al,1CH              ;
        int 21H                 ;Get interrupt vector of INT 1CH

        mov word ptr old_vect1C,bx    ;Save interrupt vector (IP)
        mov word ptr old_vect1C+2,es  ;Save interrupt vector (CS)

        mov ah,25H              ;
        mov al,1CH              ;
        mov dx,offset check_time ;Set interrupt vector of INT 1CH to new
        int 21H                 ;routine (Check_time)

        jmp no_para1
No_para0:
        mov stat_time,0FFH
No_para1:
;.....................................................................
        mov ah,25H              ;
        mov al,16H              ;
        mov dx,offset password  ;Set interrupt vector of INT 16H to new
        int 21H                 ;routine (Password)
;.....................................................................
        mov dx,offset install   ;
        int 27H                 ;Quit/ and routine stays resident
;.....................................................................
Show:   DB '╔═══════════════════════════════════╗',10,13
        DB '║ ---- FF.COM ----     Version 1.000 ',10,13
        DB '║                  IBM-PC/XT/AT/COMPAT ',10,13
        DB '║                                    ',10,13
        DB '║                    Coe & Soft Group ',10,13
        DB '║                          Udorn tech ',10,13
        DB '║ Date : 10-10-90    Time : 17:28:00 ',10,13
        DB '╚═══════════════════════════════════╝',10,13
        DB '$'

Ready:  DB '******** FF.COM already...!!',07,10,13
        DB '$'
;.....................................................................
Cseg    Ends
        End  start
;-------------------------------------------------------------------
```